

**IN THE CLAIMS**

Please amend the claims as follows:

1-2. (cancelled)

3. (currently amended) The method ~~of as claimed in claim 142~~, wherein ~~the step of~~ storing data variations is performed using a pointer.

4. (currently amended) The method ~~of as claimed in claim 14~~, further comprising, before presenting the reconstructed sequence to a user, ~~the step of:~~

~~identifying, among said identified initial and additional replets, certain~~ replets that can be used to represent multiple subsequences ~~replets~~.

5. (currently amended) The method ~~of as claimed in claim 14~~, further comprising, before presenting the reconstructed sequence to a user, ~~the step of:~~

segmenting the ~~certain~~ matching subsequences ~~replets~~ into multiple parts to account for location-specific variations of the matching subsequences ~~certain replets~~ in the sequence data.

6. (currently amended) The method ~~of as claimed in claim 14~~, further comprising, before presenting the reconstructed sequence to a user, ~~the step of:~~

storing replet information in a replet-information table using a pointer, so that equivalent replet sequences occupy single storage space.

7- 9. (cancelled)

10. (currently amended) The method ~~of as claimed in claim 14~~, further comprising, before presenting the reconstructed sequence to a user, the step of:

storing multiple views of the sequence data at multiple levels of abstraction.

11-13. (canceled)

14. (new) A computer-implemented method for storing and presenting sequence data, comprising:

specifying a set of one or more initial reptets;

for each initial reptet in the set, comparing an initial reptet to a sequence to determine a subsequence of the sequence that matches the initial reptet, if any;

generating and storing entries of match-set data structures, one match-set data structure for each reptet, a match-set data structure comprising a sequence identification to identify a sequence where a match occurs and offset information to determine a position within the sequence where the matching subsequence of the sequence is located, and wherein redundant match-set data structures corresponding to subsequences being matched by more than one reptet are not generated;

storing one or more variations for each of one or more matching subsequences, wherein a variation comprises a character in a subsequence that corresponds to a "don't care" character in a reptet that matches the subsequence;

deleting each matching subsequence from the sequence where it is found;

concatenating, in order, unmatched regions of sequences that remain after deleting each matching subsequence to form and store a backbone sequence;

reconstructing a sequence from the stored match-set data, the stored variations, and the stored backbone sequence; and

presenting the reconstructed sequence to a user of the computer.

15. (new) The method of claim 14, wherein, prior to presenting the reconstructed sequence to the user, variations are stored in a list data structure comprising a variation identification.

16. (new) The method of claim 15, wherein the list data structure comprises a subsequence character that matches a “don’t care” character in a replet that matches the subsequence.

17. (new) The method of claim 16, wherein, prior to presenting the reconstructed sequence to the user, the position of the subsequence character within the subsequence is stored in the list data structure.

18. (new) The method of claim 15, wherein an indirection pointer points to a variation so that variations common to more than one subsequence are not stored more than once.

19. (new) The method of claim 14, wherein a reconstructed sequence presented to the user is in response to a query by the user.

20. (new) The method of claim 19, wherein a query specifies a replet.

21. (new) The method of claim 14, wherein specifying a set of one or more initial reptlets comprises adding a new replet to a pre-existing set for which the steps of claim 14 have been performed.

22. (new) The method of claim 14, wherein specifying a set of one or more initial reptlets comprises removing a replet from a pre-existing set for which the steps of claim 14 have been performed.

23. (new) The method of claim 14, wherein offset information comprises two parameters,  $k$  and  $\delta$ , where  $k$  denotes a location in the sequence and  $\delta$  denotes an offset from the location.